Metody Numeryczne

Projekt 2 – Układy równań liniowych

Paulina Machcińska 191677

1. Wprowadzenie

Celem projektu jest zaimplementowanie trzech wybranych algorytmów do rozwiązywania układów liniowych. Metody użyte do rozwiązywania tych równań to metody iteracyjne Jacobiego i Gaussa-Seidla oraz bezpośrednia – faktoryzacja LU. Do implementacji wykorzystałam język C++, wykresy natomiast wygenerowałam w Matlabie.

2. Analiza zadania

2.1 Zadanie A

Dla indeksu 191677 zmienne przyjmują następujące wartości: a1 = 11, N = 977. Wektor b otrzymuję z następującego wzoru: b = sin(n*(f+1)), gdzie f = 6.

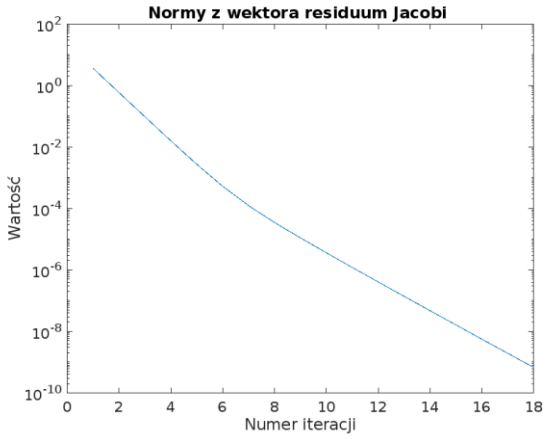
2.2 Zadanie B

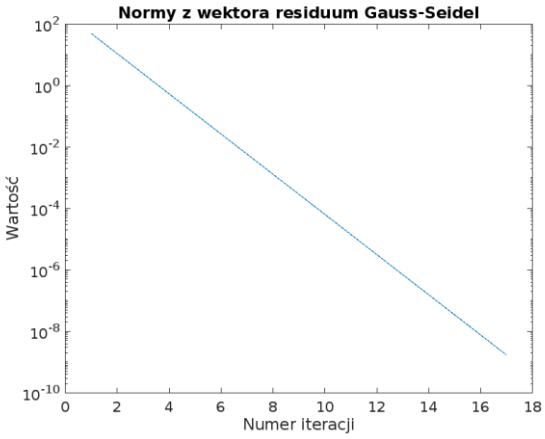
Zadanie polegało na zaimplementowaniu dwóch metod: Jacobiego i Gaussa-Seidla oraz na przetestowaniu ich działania na układzie równań z zadania A. Poniższe wyniki pokazują ile potrzebnych było iteracji, aby osiągnąć normę z wektora residuum równą 10^-9.

```
Zadanie B
Jacobi method:
Number of iterations: 13
Elapsed time: 332 milliseconds
Gauss-Seidel method:
Number of iterations: 10
Elapsed time: 230 milliseconds
```

Można zauważyć, że metoda Gaussa jest nieznacznie lepsza od metody Jacobiego. Wynik został uzyskany po mniejszej ilości iteracji oraz w krótszym czasie.

Na wykresach widać jak po każdej iteracji zmieniała się norma residuum.

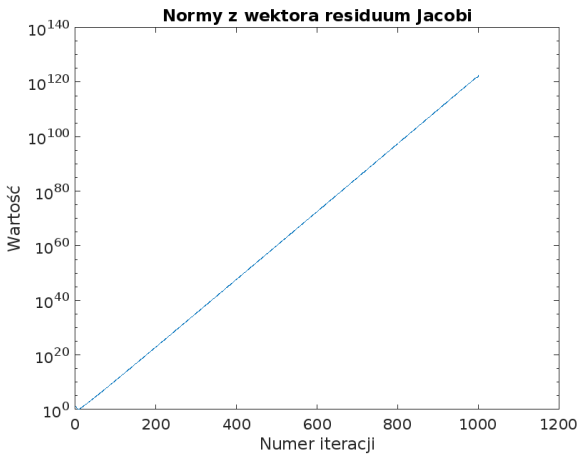


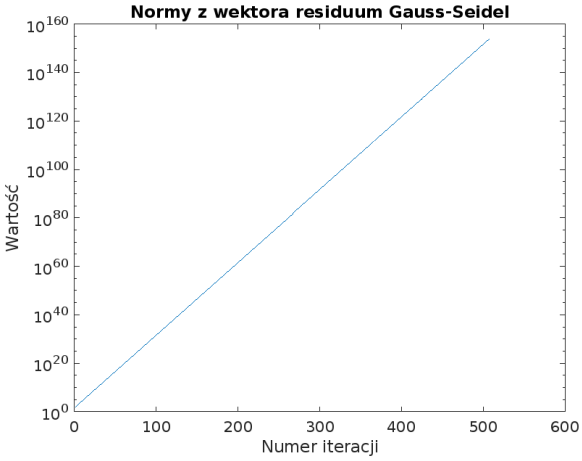


2.3 Zadanie C

W tym zadaniu należało stworzyć nieco inny układ równań i ponownie przetestować go dla tych dwóch metod. Zauważyłam, że dla tego układu obie te metody iteracyjne nie zbiegają się do dokładnego rozwiązania, więc wprowadziłam ograniczenie do maksymalnie 1000 iteracji. Bez tego ograniczenia wyskakiwał błąd wskazujący na przepełnienie pamięci.

Na wykresach widać, że wartości zamiast dążyć do 0 to rosną do nieskończoności. Można też zauważyć, że norma z residuum rośnie zdecydowanie szybciej dla metody Gaussa-Seidla.





2.3 Zadanie D

W tym zadaniu należało zaimplementować metodę bezpośrednią, która w przeciwieństwie do metod iteracyjnych zawsze znajduje dokładne rozwiązanie układu równań liniowych. Dla układu równań z zadania C algorytm uzyskał następujące wyniki:

```
Zadanie D
LU method:
Residuum norm: 1.99827e-12
Elapsed time: 1.5063 milliseconds
```

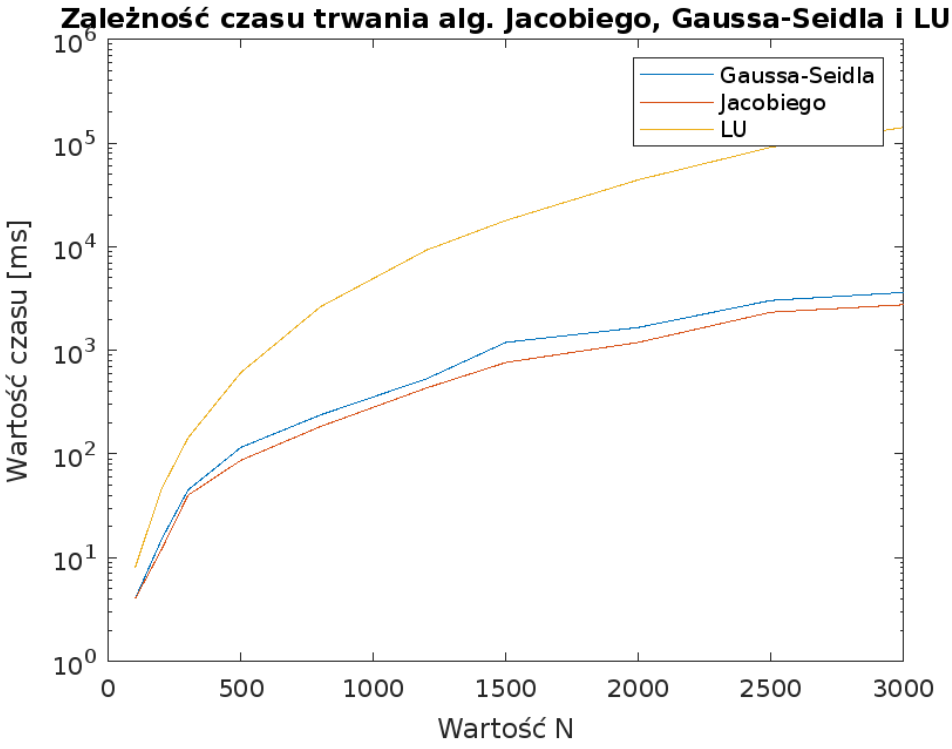
Norma z residuum jest bardzo bliska 0, co oznacza wysoką dokładność wykonanych obliczeń.

2.3 Zadanie E

W tym zadaniu należało stworzyć wykres przedstawiający zależności czasu trwania poszczególnych algorytmów od liczby niewiadomych N={100, 200, 300, 500, 800, 1200, 1500, 2000, 2500, 3000} dla układu równań z zadania A.

Jak można było się spodziewać, czas wykonywania algorytmów w zależności od liczby niewiadomych rośnie wraz ze wzrostem N dla każdego z algorytmów. Metoda Jacobiego oraz Gaussa-Seidla mają porównywalne wyniki, jednak w zależności od N druga metoda jest minimalnie szybsza. Natomiast faktoryzacja LU ma znacznie większe czasy oraz szybszy ich wzrost od metod iteracyjnych, szczególnie dla większych macierzy.

3. Wnioski



Z analizy przeprowadzonej w ramach tego projektu wynika, że metoda faktoryzacji LU wymaga więcej czasu obliczeniowego niż iteracyjne, jednak zawsze znajduje dokładne rozwiązanie, co czyni ją lepszą opcją w przypadkach, gdy dokładność jest priorytetem. W przypadku metod iteracyjnych, ich zaletą jest szybkość działania, jednak nie zawsze są one w stanie znaleźć dokładne rozwiązanie dla niektórych układów równań – co mogliśmy zaobserwować w zadaniu C.

Bezpośrednia metoda rozwiązywania układów równań liniowych pomimo dłuższego czasu działania, zawsze znajdzie rozwiązanie. W związku z tym, w przypadku, gdy metody iteracyjne nie są wystarczające do znalezienia dokładnego rozwiązania, należy zastosować metodę bezpośrednią.